

## REMARKS

### I. Office Action Summary

Claims 24, 38 and 45 are the pending independent claims. In the Office Action dated August 22, 2005, the Examiner rejected claims 24-26, 29-38, and 40-45, under 35 U.S.C. §102(e) as being anticipated by Lumelsky et al (U.S. Patent No. 6,466,980). The Examiner also rejected claims 27-28 and 39 under 35 U.S.C. §103(a) as obvious over the combination of Lumelsky et al. in view of Bowman-Amuah (U.S. Patent No. 6,578,068).

### II. Rejections Under 35 U.S.C. § 102(e)

Applicants respectfully disagree with the Examiner's rejections of claims 24-26, 29-38 and 40-45 over U.S. Patent No. 6,466,980 (Lumelsky). As discussed in greater detail below, Lumelsky discloses using location information in service of optimizing **distribution of multimedia objects**<sup>1</sup>, whereas Applicants' specification **manages location information** separately from the data to which the location information pertains<sup>2</sup>.

#### CLAIM 24

Claim 24 relates to a method of scaling at least one of location server capacity and transaction rate capability in a system for storing and retrieving location information. Claim 24 includes, *inter alia*, the step of:

---

<sup>1</sup> See: "A system and method for dynamically shaping available capacity of **multimedia objects** based on aggregated demand across **distributed media/world-wide-web servers**" (Abstract; **bold added**). See also: "Thus, unlike the prior art system 10, the various **collections of objects (130, 131, 132)** found at each independent server (**1201, 1211, 1221**) aggregate now into a **distributed collection (130, 131, 132) of objects** and **object replicas** which model a **multimedia object** as a **scalable** and relocatable resource **in accordance with demand and capacity conditions**" (col. 8, ll. 40-46; **bold added**).

<sup>2</sup> See specification: "According to one aspect of the invention, a system for managing **location information** and providing **location information** to data locations queries comprises a transfer protocol configured to manipulate an identifier, and at least one location associated with the identifier, wherein the identifier uniquely specifies an entity and wherein each data location specifies a location of data in a network pertaining to the entity" (Applicant's p. 3, ll. 4-9; **bold added**). **See also**: "A network distributed tracking protocol (NDTP) is a transfer protocol having the capability to manipulate **location information** used to efficiently track the **location of information** associated with an individual entity in the distributed database system" (Applicant's p. 6, ll. 14-17; **bold added**).

transferring a portion of the identifiers and associated locations to a second data location server when a performance criterion of the first server reaches a predetermined performance limit (p. 46, ll. 25-27).

Thus, claim 24 relates to improvements for transaction rate scalability for **location information** separate and apart from any data to which the location information points.<sup>3</sup> As recited in claim 24, transferring a portion of the identifiers and associated locations to a second **data location server** involves isolating specifically the location information from the end-point data itself and making changes to how the location information is managed, not the end-point data that is stored at a location identified by the location information. Examples of partitioning portions of the **location information** across servers are found in Applicants' specification. Applicants' specification includes techniques and network topologies for optimizing transaction rate including [1] clustering and [2] distributed topology replication of *associated identifier and location information*.

For example, FIG. 5 illustrates a flat NDTP server topology using [1] **clustering**, or a [2] **distributed topology using replication**, where each of the NDTP servers 12 in the cluster may contain a **different portion of a pool of associated identifier and location information** (p. 11, ll. 30 – p. 12, l. 1; **bold** and *[enumeration]* added).

Clustering and distributed topologies using replication separate specifically location information onto different machines, thereby allowing portions of the location information to be processed by more server and network resources.

In contrast to Claim 24, Lumelsky only focuses on replication of data objects rather than on management of specifically location information separated from the data objects themselves. Lumelsky teaches the use of location information to improve transmission performance and scalability of multimedia objects.<sup>4</sup> Instead of teaching

---

<sup>3</sup> Regarding (a) **transaction rate**, see: "An advantage of distributing an NDTP server data set across independent machines is that **both capacity and transaction rate scale can be increased**. In one embodiment, each additional machine in an NDTP server cluster 50 **linearly increases capacity**, by adding main and secondary storage, and **transaction rate**, by adding processing power and network bandwidth (assuming a properly scalable network infrastructure is employed)" (p. 25, ll. 24-30; **bold** added). Regarding (b) **specifically location information**, see: "For example, FIG. 5 illustrates a flat NDTP server topology using clustering, or a distributed topology using replication, where each of the NDTP servers 12 in the cluster may contain a **different portion of a pool of associated identifier and location information**" (p. 11, ll. 30 – p. 12, l. 1; **bold** added).

<sup>4</sup> See note 1.

the management of location data itself so that data location servers with data location information are reconfigured to handle queries for the location of data, Lumelsky discloses managing and replicating the end-point data itself (in Lumelsky, the data objects disclosed are multimedia files).<sup>5</sup> The method of claim 24 relates to the management of specifically location information and capacity and transaction rate scaling for accessing the location information needed to access data, while Lumelsky is concerned with the replicating and managing the data itself rather than the location information relating to the whereabouts of the data.

The Examiner asserts that Lumelsky teaches or suggests transferring portions of identifier-location associations to a second server given a performance limit. On page 3, the Examiner asserts that Lumelsky teaches:

transferring a **portion of the identifiers and associated locations** to a second data location server when a performance criterion of the first location server reaches a predetermined performance limit (Examiner remarks, p. 3; **bold added**).

In support of this assertion, the Examiner cites Lumelsky:

More particularly, the present invention introduces the notion of a global server which **provides a spare, shared, and highly available capacity that can be used to assist a multimedia server** by temporarily increasing the overall system capacity associated **with some particular multimedia object** to match its predicted demand. The present invention additionally introduces the notion of a **transient replica** which replica acts as a **migrating object** of limited lifetime that responds to demand and capacity conditions. To do so, the controller node monitors demand and capacity and uses, creates, and deletes **transient replicas** from global servers (col. 6, ll. 33-43; **bold added**).

Lumelsky's "spare, shared, and highly available capacity" refers to resources assisting multimedia objects not location information. Lumelsky's "transient replica" refers to a multimedia object not location information. Therefore, Lumelsky teaches transferring data objects to additional servers but does not teach transferring portions of location information to additional servers. Lumelsky fails to teach or suggest the separate management of location information pertaining to the data, therefore Lumelsky does not teach transferring a portion of the identifiers and associated locations to a

---

<sup>5</sup> See: "Large databases store large amounts of data, but the NDTP server is only concerned with the location of that data, **rather than the end-point referent data itself**" (p. 8, ll. 23-26; **bold added**). "End-point referent data itself" are multimedia objects in Lumelsky.

second data location server when a performance criterion of the first location server reaches a predetermined performance limit as claimed in claim 24.

Thus, Lumelsky does not teach either (a) a method of scaling at least one of location server capacity and transaction rate capability in a system for storing and retrieving location information, or (b) transferring a portion of the identifiers and associated locations to a second data location server when a performance criterion of the first location server reaches a predetermined performance limit. For at least these reasons, Applicants submit that independent Claim 24 is patentable over Lumelsky. Claims 25-37 are dependent claims, therefore their allowability directly follows from the allowability of independent claim 24.

#### CLAIM 38.

Claim 38 relates to a database using an indexing function to map identifiers and index designations. Claim 38 includes, *inter alia*, the step of:

an indexing function stored in the computer readable medium, the indexing function operative to map each of the plurality of identifiers to a respective one of the plurality of index designations

An example of an indexing function may be found in Applicants' specification at, for example p. 27, ll. 15-22, where a well-known function for mapping identifiers to index designations is described:

The presently preferred embodiment for accomplishing this [NDTP database insertion and scaling] is to define **a well-known function of the identifier** and **use this function** to select from a set of NDTP servers. NDTP clients will preferably apply this well-known function to the identifier for each NDTP request, and send the NDTP request to the indicated NDTP server. This technique will effectively partition the set of all identifier/location associations across the NDTP servers. **Each NDTP server will only maintain the portion of the total association set which corresponds to its particular identifiers** (bold added).

An example of a "well-known function" is a hash function.<sup>6</sup> Using a specific **indexing function** Applicants deliberately permit distributing portions of the database's location information onto separate servers. In contrast, Lumelsky teaches three **components** (but not a function) to interact with placement recommendations and inquiries:

---

<sup>6</sup> Applicants' specification (p. 28, ll. 6-27) provides an example algorithm and specific usage conditions.

Particularly, the placement module (610) **interfaces with** both a replica directory service (665) (for maintaining the replica directory 666 as described herein with respect to FIG. 6(a)) and a server directory service (655) (for maintaining the server directory 656 as described herein with respect to FIG. 6(b)) **to generate the tentative placements** (620). That is, the placement module (610), replica directory service (665), and replica directory (666) **operate in conjunction** to locate all replicas associated with the given object identifier of the received request. Further, the placement module (610), server directory service (655) and server directory (656) **operate in conjunction** to determine the willingness of any such location (holding a replica) to consider new placement inquiries (620) (Col. 9, ll. 31-44; **bold added**).

Lumelsky teaches that the (a) placement module, (b) replica directory service, and (c) server directory service interact to generate placement recommendations and to consider placement inquiries, but Lumelsky does not teach an indexing function. Therefore, not teaching an indexing function, Lumelsky does not teach an indexing function operative to map each of the plurality of identifiers to a respective one of the plurality of index designations.

Illustrated in FIG. 6(a) and FIG. 6(b), Lumelsky further specifies the replica and server directories.<sup>7</sup> The replica directory records various statistics for identifiers, such as predicted demand, geography, demand volume, and time-to-live (col. 10, ll. 23-28). The server directory records information such as IP address or hostname and capacity rating (col. 10, ll. 42-46). These are values. However, Lumelsky does not teach an indexing function to map identifiers to index designations. Thus, Lumelsky does not teach an indexing function operative to map each of the plurality of identifiers to a respective one of the plurality of index designations.

Therefore, while Lumelsky teaches a placement module interacting with replica and server directory services, Lumelsky does not teach a database comprising the use of an indexing function stored in the computer readable medium, the indexing function operative to map each of the plurality of identifiers to a respective one of the plurality of index designations. For at least these reasons independent Claim 38 is patentable over Lumelsky. Claims 39-44 are dependent claims, therefore their allowability directly follows from the allowability of independent claim 38.

---

<sup>7</sup> See discussion of replica directory in Lumelsky, Col. 10, ll. 21-35, and server directory in Col. 10, ll. 36-62.

## CLAIM 45

Claim 45 relates to a system for managing location information and providing location information to location queries. Claim 45 includes, *inter alia*, the step of:

a location server operating in accordance with a transfer protocol, the transfer protocol comprising instructions for manipulating an identifier and at least one location associated with the identifier

Different than Lumelsky, claim 45 recites a transfer protocol for **specifically managing location information**. Applicants' specification provides detail and figures describing an example of suitable protocol instructions for database management of specifically location information.<sup>8</sup>

A network distributed tracking protocol (NDTP) is a transfer protocol having the capability to manipulate **location information** used to efficiently track the location of information associated with an individual entity in the distributed database system (p. 6, ll. 14-17; **bold added**).<sup>9</sup>

The NDTP transfer protocol runs on standard lower-level network transport protocols, such as TCP and UDP:

NDTP defines a transaction oriented protocol, which can be carried over any of a variety of lower level network transport protocols. TCP and UDP are currently supported, however any of a number of other protocols are also supportable (p. 21, ll. 6-9; **bold added**).

NDTP is carried over TCP and UDP transport protocols, and NDTP is a transfer protocol designed for managing location information. In contrast, Lumelsky teaches use of transfer and transport protocols for transmitting multimedia data rather than for managing location information:

---

<sup>8</sup> See Applicants' specification, FIGS. 8-17B (p.12, l. 7 – p. 20, l. 20), which provides examples of instructions for database interaction and management of location information. Example instructions for location information include: (a) querying the database for identifier associations (NDTP\_GET, NDTP\_GET\_RSP, NDTP\_GET\_TRANSACTION), (b) adding location associations to an identifier (NDTP\_PUT, NDTP\_PUT\_RSP, NDTP\_PUT\_TRANSACTION), (c) deleting entries from the database (NDTP\_DEL, NDTP\_DEL\_RSP, NDTP\_DEL\_TRANSACTION), (d) updating records in the database (NDTP\_UPD, NDTP\_UPD\_RSP), and redirecting queries (NDTP\_RDR\_RSP).

<sup>9</sup> See also applicant specification: "The Network Distributed Tracking Protocol (NDTP) efficiently tracks the **location of data** associated with an individual entity in a distributed database" (p. 12, ll. 8-9, **bold added**).

It should be noted that standards for controlling multimedia streaming data over the World Wide Web such as H. 323 and Real Time Streaming Protocol (RTSP) are already in place and implemented to provide the streaming capabilities they are intended for. Whereas H.323 is **designed for videoconferencing** across small groups, RTSP is designed to efficiently **broadcast audio-visual data** to large groups. Each standard [H.323 and RTSP] describes a client-server application-level protocol for **controlling the delivery of data** with real-time properties. For example, the RTSP establishes and controls either a single or several time-synchronized **streams of continuous media**, such as **audio and video** and **uses transport protocols** such as **UDP, multicast UDP, TCP, and Real Time Protocol (RTP)** to **deliver the continuous streams** (Lumelsky, Col. 9, ll. 14-21; **bold added**).

Lumelsky teaches the use of transfer protocols designed for (a) videoconferencing and (b) broadcasting audio-visual data, (c) controlling delivery of data, and (d) lower-level transport protocols UDP, multicast UDP, TCP, and RTP to deliver continuous streams. However, Lumelsky does not teach use of a transfer protocol comprising instructions for manipulating an identifier and at least one location associated with the identifier.

Lumelsky additionally teaches use of a controller device for managing placement of multimedia objects:

FIG. 5 is a detailed block diagram of the intermediary **controller device (520)** implemented for **managing the placement of multimedia objects themselves** onto servers. As shown in FIG. 5, a request processing module (600) is provided for receiving requests (**601, 602, 603, 604**) from clients, the request including a unique object identifier, and feeding these requests to a **placement module (610)**. The placement module (610) applies a **placement policy (615)** to each request and generates a set of tentative placement queries (**620**) for the request. Particularly, the placement module (610) **interfaces with both a replica directory service (665)** (for maintaining the replica directory **666** as described herein with respect to FIG. 6(a)) and a **server directory service (655)** (for maintaining the server directory **656** as described herein with respect to FIG. 6(b)) to generate the tentative placements (**620**) (Col. 9, ll. 22-37; **bold italic added**).

Lumelsky teaches use of a controller device for managing the placement of multimedia objects, using a placement module within a controller device, applying a placement policy, and interfacing the placement module with replica and server directory services, to generate tentative placements. However, Lumelsky does not teach the use of a transfer protocol comprising instructions for manipulating an identifier. Therefore, Lumelsky does not teach a location server operating in accordance with a transfer

protocol, the transfer protocol comprising instructions for manipulating an identifier and at least one location associated with the identifier.

Finally, Lumelsky teaches the use of replica and server directories to advise placement policy, illustrated in FIG. 6(a) and FIG. 6(b).<sup>10</sup> The *replica directory* records various statistics for identifiers, such as predicted demand, geography, demand volume, and time-to-live (col. 10, ll. 23-28). The *server directory* records information such as IP address or hostname and capacity rating (col. 10, ll. 42-46). However, Lumelsky does not teach the use of a *transfer protocol* comprising instructions for manipulating an identifier and at least one location associated with the identifier.

Lumelsky teaches the use of transport protocols for transmitting multimedia data, including videoconferencing, audio-visual data, and continuously streaming data. Lumelsky also teaches using a controller device and directory services for replicas and servers relating to placement policy of multimedia objects. However, Lumelsky does not teach the use of a transfer protocol comprising instructions for manipulating an identifier and at least one location associated with the identifier. For at least these reasons Applicants submit that claim 45 is patentable over Lumelsky.

#### **IV. Rejections Under 35 U.S.C. § 103(a)**

Applicants respectfully disagree with the Examiner's rejections under 35 U.S.C. § 103. Applicants submit that dependent claims 27-28 and 39 are allowable for at least the same reasons as provided above for their respective independent claims.

---

<sup>10</sup> See discussion of replica directory in Lumelsky, Col. 10, ll. 21-35, and server directory in Col. 10, ll. 36-62.

**V. Conclusion**

In light of the above, Applicants submit that claims 24-45 are in condition for allowance. If any issues arise or questions remain, the Examiner is invited to contact the undersigned at the number listed below in order to expedite disposition of this case.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Kent E. Genin', is written over a horizontal line.

Kent E. Genin  
Registration No. 37,834  
Attorney for Applicants

BRINKS HOFER GILSON & LIONE  
P.O. BOX 10395  
CHICAGO, ILLINOIS 60610  
(312) 321-4200